

N64 13916 *

CODE-1

(NASA CR-55267)

40p.

SELF-REPAIR TECHNIQUES

FOR FAILURE FREE SYSTEMS

T 2 Special Technical Report No. 2

UNPUBLISHED PRELIMINARY DATA

(NASA Contract Nasw-572)
Reference WGD-38521

M. R. Coagrove and
C. G. Masters
September 1963 40p ref

OTS: PRICE

XEROX

\$

3.60 ph.

MICROFILM

\$

1.40 mf.



9431008

Westinghouse Electric Corporation Baltimore, Md.

Electronics Division

P. O. Box 1897

Baltimore 3, Md.

Special Technical Report No. 2

On

Self-Repair Techniques
For Failure-Free Systems

Contract Nasw - 572

Reference WGD-38521

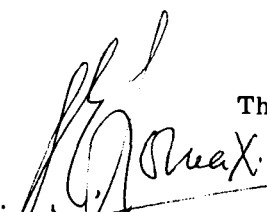
by

M. R. Cosgrove

C. G. Masters

September 1963

APPROVED:


S. E. Lomax, Director
Advanced Development Engrg.

The Westinghouse Electric Corporation
Electronics Division
Box 1897, Baltimore 3, Maryland

TPE 5059

13916

ABSTRACT

This report describes the initial step in the design of an optimal self-repairing system. The report contains a description of the several classes of "repair" strategies under consideration and the computer simulation program which is used to determine the performance of the systems for each strategy.

The computer simulation program determines the performance of a particular strategy by injecting random failures throughout the system and simulating system reaction according to the "repair" pattern of the strategy in question. The program prints out system performance in terms of:

1. total time to failure
2. average time to failure
3. number of failures to system failure
4. number of switches affected.

The results for the two classes of strategies for which curves were drawn show that with the addition of a minimal amount of self-repair capability, the reliability of the system can be substantially increased over that of a comparable system using fixed redundancy alone for failure protection.

AUTHOR

TABLE OF CONTENTS

	Page
ABSTRACT.	ii
I. INTRODUCTION	1
II. STRATEGY DESCRIPTION	5
A. Basic Assumptions.	5
B. Basic Strategy Classes Considered to Date	5
III. THE COMPUTER SIMULATION PROGRAM	9
A. The Reason a Simulation Program was Used	9
B. How the Program Works	9
C. Sample Format	12
D. Production Format	13
IV. RESULTS	15
A. Failures Withstood (as percent of system) vs. Spare Mobility . . .	15
B. Reliability vs. Time Curves	17
V. SUMMARY AND CONCLUSIONS.	25
VI. FUTURE STUDIES	27
VII. APPENDIX	29

LIST OF ILLUSTRATIONS

<u>Figure</u>		<u>Page</u>
1	Multiple-line Redundant System.	2
2	Multiple-line Redundant System with Self-Repair Capability	2
3	Probability Distribution of a Component Failure	10
4	Simulation Matrix	12
5	Average Number of Failures Withstood (As Percent of Gamma 1 Systems) Versus Number of Moves per Spare.	16
6	Average Number of Failures withstood (as Percent of Beta Systems) Versus Number of Spares per Block	18
7	Minimum Number of Failures (as Percent of Gamma 1 Systems) Versus Number of Moves per Spare	19
8	Minimum number of Failures (as Percent of Beta Systems) Versus Number of Spares per Block	20
9	Percent of Systems Operating (Beta Class) Versus Time	22
10	Percent of Systems Operating (Gamma Class 1) Versus Time	23

I - INTRODUCTION

In an effort to increase the reliability of complex electronic systems, several methods have been proposed for using "redundant" equipment to provide failure protection within these systems. Two of the most useful types of redundancy techniques are multiple-line, majority voted logic and multiple component grouping schemes. Although both techniques are very effective, a large percentage of the "redundant" equipment is not efficiently used, i. e., the system fails with much of the "redundant" equipment still functioning. This undesirable feature is inherent in systems of this type because random failures do not tend to distribute evenly throughout the system. Instead, they almost invariably tend to group and cause a critical failure pattern to occur in one subsystem area before many failures have occurred in the remainder of the system. The most drastic example of this is the failure of an order three, multiple-line, majority voted system upon the occurrence of two successive failures in the same stage with no other failures in the remaining stages.

Westinghouse has devised a new solution to the failure protection problem which exploits most of the desirable features of the multiple-line, majority-voted schemes, but is not as sensitive to critical failure patterns as the more standard techniques. This solution is in the form of a set of strategies for allowing the reorganization of the systems in response to failure patterns which may develop. The systems which employ these strategies are called self-repairing systems.

The general approach of the self-repair strategies can be described through the use of an example. Figure 1 shows a block diagram of an order three, multiple-line system. Figure 2 shows the same system after some self-repair capability has been added. It is assumed that all blocks in the system are functionally identical such as the multivibrators in a shift register, and are interconnected by switching and voting circuits. If two blocks in the same column fail and the blocks on either side of this column are still operating, the self-repair switching mechanism senses this condition and shifts the required additional working blocks to the failed column. The failed block can now be eliminated or "voted out." This procedure decreases the remaining protection provided the adjacent columns, but it prevents system failure at a critical point and thus extends the life of the system. As additional blocks fail, other blocks are switched into the failed columns. The choice of which block shall be brought in to aid the vulnerable column is determined by the particular strategy in use.

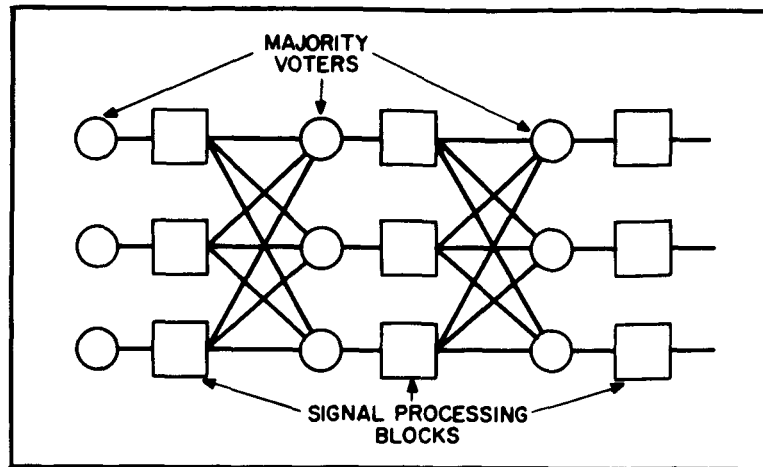


Figure 1. Multiple-line Redundant System

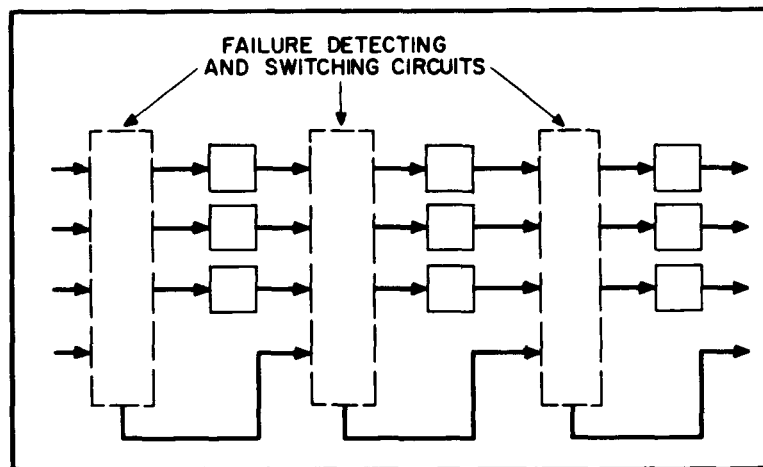


Figure 2. Multiple-line Redundant System with Self-repair Capability

The unique feature of these strategies is that the switching circuitry can be completely distributed rather than "lumped" into a central controller. As a result, most failures in the switching circuitry are equivalent to signal processor (block) failures and are eliminated in the normal manner. This means that individual failures in the switching circuitry do not cause the loss of the entire self-repair capability.

Before a "hardware" design of self-repairing systems can begin, the full range of feasible switching strategies must be examined, and from these an optimum strategy or set of near optimum strategies must be selected. The majority of this report is concerned with

a description of some of the more promising strategies and with the computer program which is being used to simulate the failure response of systems which employ these strategies.

There are a great number of possible strategies which may be investigated, many of which are quite similar to one another. The strategies being considered are arranged in groups called classes, the individual members of which are special cases of the general class. This allows the investigation and programming of a few classes of strategies rather than many individual strategies. This facilitates comparison of strategies within a class as well as adding a certain degree of generality to the analysis.

Before proceeding to the description of specific strategies or classes of strategies, the properties a self-repairing system should have must be noted and the basic assumptions stated. A short list of the general desirable properties is compiled below.

- a. Self-repairing systems should be more reliable than ordinary redundant systems of identical function capability and cost.
- b. The switching strategy used should make optimum use of the redundant function blocks for a fixed amount of switching complexity.
- c. Instantaneous failure masking must be provided for system applications which cannot withstand a temporary loss of data. An example of this is the key-stream generator used in secure communication channels.
- d. The strategy must be suitable for implementation by a distributed (non-centralized) switching network.

II - STRATEGY DESCRIPTION

A. BASIC ASSUMPTIONS

Almost all large computing and control systems are formed by interconnecting a relatively small number of different types of basic circuit blocks. As a result, the components of these systems can be split up into homogeneous groups of functionally similar or identical blocks. It is assumed, therefore, that such groups can be formed and that self-repair strategies can be applied within each group. Note: The members of any group are not required to be physically or functionally adjacent but may be located in scattered sections of the overall system.

It is also assumed that at least two blocks must be performing the same nominal function before a failure can be detected, and at least two correctly operating blocks must be performing the same function before a third (failed) block can be eliminated from this function.

If at least three blocks are performing a function and one of them fails, the elimination process is assumed to be instantaneous, and the failure is assumed to be completely masked. If, however, only two blocks are performing the function and one fails, a third block must be switched to that location to eliminate the failure. This process is not assumed to be instantaneous and errors appear in the system temporarily. As a result, systems using the basic order-three redundancy with self-repair (as will be described in the Beta and Gamma Class strategies of this report) must be capable of withstanding temporary data loss without mission failure. If this assumption is not true, a higher order of redundancy must be used as in the Alpha class strategies or higher-order versions of the Beta and Gamma classes.

If, because of particular failure and response patterns, single blocks are left to perform particular functions it is assumed that the system continues to operate with one or more stages existing in the non-redundant state either until one of these blocks fails or until another critical failure pattern occurs elsewhere in the system.

Finally, it is assumed that a stage shown pictorially at one end of a system is, in reality, adjacent to the opposite end and enjoys the same repair facilities as stages shown in the center of the system.

B. BASIC STRATEGY CLASSES CONSIDERED TO DATE

The following few paragraphs will indicate the general principles of each of the three strategy classes which have been simulated thus far. Detailed examples of each class are shown in the Appendix, and the reader will probably need to refer to these for detailed consideration of the following descriptions.

1. Alpha (α) Class

Systems employing the α class strategies are basically multiple-line redundant (usually order three) systems which are equipped with sets of spares. These spares are additional function blocks which can be automatically used to replace failed blocks. In general, spares can not economically be given enough mobility to allow a single spare to be capable of replacing each operational block in the entire system. Instead, individual spares are usually given restricted capability and may replace only blocks in a single row* or portion of a row. A large number of strategies, each belonging to the (α) class, can be generated by varying (a) the total number of spares available for a fixed system size, (b) the mobility of each spare (c) the pattern in which the spares' repair capabilities overlap.

If it is assumed that spares will immediately replace failed blocks regardless of whether it is the first failure in a function column or not, complete failure masking is achieved. The threshold vote technique will continue to absorb failures after the spares complement is exhausted until a majority of unrepairable failures have occurred at a particular function. At this point the system will fail since both the self repair capability and the network redundancy have been exhausted.

2. Beta (β) Class

Beta Class strategies do not utilize inactive spare blocks as does Class α . With no failures, the system operates as an ordinary multiple-line redundant system. When a critical failure i. e., one which would cause failure of a multiple-line redundant system, occurs, the failed block is removed from the system and replaced by a properly functioning block from an immediately adjacent function. The individual strategies in this class differ from one another primarily in the number of spares which they can draw from the rest of the system.

Because failures are replaced by function blocks only from the adjacent functions there is a smaller amount of switching circuitry involved with Class β than with other classes of self-repair strategies. This advantage is partially offset, however, by the one drawback inherent in this class of strategies. That is these systems are more vulnerable to failures which are grouped in one area of the system than are the more flexible strategies.

The three strategies of this type which have been simulated are described in the Appendix. These particular strategies do not usually allow blocks to move a second time after an initial repair has been made. This restriction has been made for a variety of reasons, but other strategies are being considered which will release this restriction. In addition, strategies having increased spare mobility will be considered in future studies.

* For example the top line or row of signal processor in Figure 1.

3. Gamma (γ) Class

The Gamma (γ) Class of self-repair strategies contains much more variety than either Class α or Class β . The class is characterized by a shifting of the spare blocks in one direction to alleviate the critical condition caused by the failed function blocks. Unlike the strategies of Class β , it is possible for a spare to move several times in response to failures. When a critical failure occurs, one of the function blocks adjacent to the failure will replace it, leaving a void. This void, if it creates a vulnerable situation i. e., one block per function stage, will be filled by the function block immediately adjacent to it in the opposite direction from the original failure. The next failure to occur in the same stage as the original failure causes another shift of the function block now adjacent to the failure. This may be a function which has already shifted in response to a failure. As long as spares are available, they will continue to shift laterally to replace failed blocks or to fill voids.

Since the spare function blocks are allowed much more mobility in this class of strategies, more failures can be corrected. However, the amount of switching circuitry necessary to implement the strategies is a monotonically non-decreasing function of the mobility of the spares. This creates problems of implementation which limit the usefulness of high spares mobility.

The individual members of Class γ strategies differ primarily in amount of mobility allowed to the function blocks. This, in turn, affects the failure absorption capabilities of the strategies. Again, the individual strategies are described in more detail in the Appendix.

III. THE COMPUTER SIMULATION PROGRAM

A. THE REASON A SIMULATION PROGRAM WAS USED

Although the reorganization features of self-repairing systems improve the failure absorption capability of redundant networks, these features drastically affect the analytical reliability expressions developed for multiple-line, majority-voted systems. Not only does a slight amount of reorganization capability greatly complicate the expressions, but each modification of each strategy class appears to require a different solution. Extensive efforts to model some of the simpler self-repairing systems have been unsuccessful. Because of this, efforts to write exact reliability expressions have been dropped, and a general computer simulation program has been written to facilitate a Monte Carlo approach to the reliability analysis. This program can be used to simulate a broad range of strategies, and it provides data about the actual switching patterns which tend to occur in a system. This latter information could not be easily determined from reliability expressions even if they were available. A plot of reliability versus time can be obtained directed from the program results with no more additional input information than would be required by calculations made using analytical expressions.

B. HOW THE PROGRAM WORKS

1. The General Program Philosophy

A redundant system of the desired order of redundancy and number of functions is set up in matrix form. The strategy class is then selected from a group of sub-programs and input data which specifies the particular strategy to be tested is read in. Through the use of a series of random numbers, individual blocks are designated as failed, and the switching strategy responds to each failure until the system fails to pass the operational criteria. A second series of exponentially distributed random numbers determines the time between each simulated failure, and the sum of these is the time to system failure. Once the system fails, the pertinent data is recorded, and the computer resets and begins to generate two new sets of random numbers. Continued repetition of this process provides the compilation of data mentioned in part A of this section. The following paragraphs indicate specifically how the various portions of the program work and the form of the print out.

2. The Failure Selection Program

A simple procedure for randomly selecting the failed function blocks has been set up. Each block is assumed to have an exponentially decaying reliability $= e^{-\lambda t}$ where λ is a constant failure rate. It has been shown that the conditional probability that a failure

has occurred in the i^{th} block given that a failure has occurred in the system is equal to the

constant, $\frac{\lambda_i}{\sum_{i=1}^N \lambda_i}$.

If the interval between zero and one is split into N subintervals, each proportional to the associated conditional probability, a set of random numbers uniformly distributed between zero and one can be used to determine which blocks fail with correct conditional probability of picking any one box. In this particular computer program, the random number specifies the block to be failed. The system then responds to eliminate the failed block. If the response is possible, i. e., a spare block is available to make the repair, a new random number is chosen and the procedure repeats. If no spare is available, the system is judged as failed.

3. Time Determination

For each of the simulated failed blocks selected above, a time to failure for the block is also determined. A. M. Mood¹ has shown that random numbers taken from the uniform distribution can be transformed into any desired continuous distribution by letting

$$f(y) = 1 \quad 0 < Y < 1$$

$$y = G(x)$$

Where $G(x)$ is the cumulative distribution of x .

This relationship is shown graphically in figure 3.

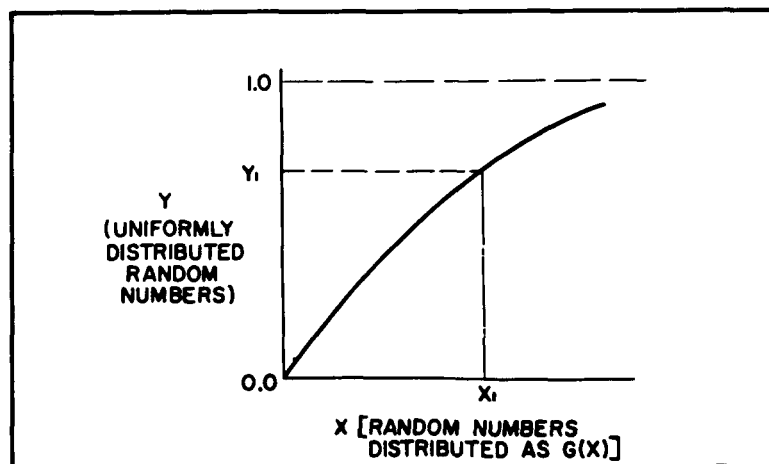


Figure 3. Probability Distribution of a Component Failure

¹Mood, A. M. - Introduction to the Theory of Statistics McGraw Hill Book Co., Inc. 1950

Y is a single valued function of x and vice versa. For each Y chosen from a uniform distribution, a unique value of x is determined.

The G (x) function which is of particular interest here is $G(t) = 1 - R(t) = 1 - e^{-\lambda t}$. This is the distribution function associated with the probability that the first failure has occurred within a system. This curve is shown in figure 3.

For the first function block failure, a random number is chosen from a uniform population and transformed to a corresponding number from the exponential distribution. This latter number is the time from system start to the first failure. To calculate the time to the second failure, the λ associated with the first failed block should be subtracted from the $\Sigma \lambda$'s and the procedure repeated. The new number thus obtained would be the time from the occurrence of the first failure to the occurrence of the second failure. When the system fails, the sum of these individual failure times will determine the total system operating time.

In the present program, the above procedure is slightly modified to make computations easier. Instead of decreasing the $\Sigma \lambda$'s after each failure, this sum is left the same and blocks are allowed to fail more than once. When a block fails for the second time no action is taken other than to add the time to this failure to the system operating time. This modified procedure would not be acceptable if the times between subsystem failures were of interest, but since total system operating time is the only factor to be considered, the results are almost identical to these which would be obtained in the more straightforward approach.

4. The System Reactions

It is obvious that many specific reactions are different for different strategies, but the general manner in which the program performs the various shifts and the type "bookkeeping" involved can be briefly described. Figure 4 schematically illustrates the form in which computer "views" the system to be simulated. The height of the "basic array" is set by the original order of redundancy, the width by the number of stages, and the depth by the number of data words associated with each block. The "failed block array" is a two-dimensional array into which the data words for failed blocks are shifted as the failures occur. The only indication to the computer that a block has failed is the shifting of these data words into this latter array.

When a set of data words is moved into this array, the computer examines the remainder of the system and makes any necessary response. This is done by shifting the data words associated with the appropriate spare blocks from their original locations into the locations specified by the particular switching strategy being considered.

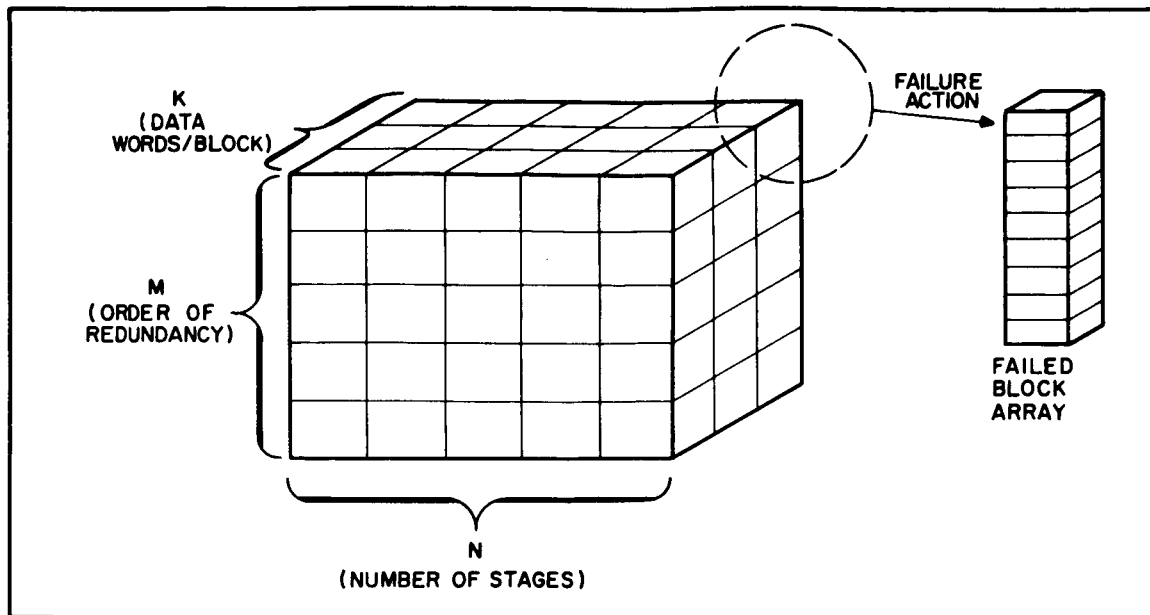


Figure 4. Simulation Matrix

C. SAMPLE FORMAT

A check must be made to determine whether the computer simulation program is operating correctly, i. e., selecting the correct function block for failure according to the random number set, responding properly to failures according to the particular strategy, and failing at the proper time and under the proper conditions. In order to accomplish this, a sample format has been developed. This sample format prints out the following information:

1. * The function block designations and the random number range which describes failure of the block.
2. * A list of failures which occur with all the information associated with the failure such as:
 - a. The random number which was selected
 - b. The location of the failed block
 - c. The amount of time from the previous failure to the time of failure of the block in question
 - d. The cumulative time from the beginning of system operation.
3. The average time between failures.

* This information is printed out for each failure until the system fails.

When a critical failure of a function block occurs, an operating spare is switched into the vacant position by assigning random number limits of the spare block to the failure location. This permits checking of the switching pattern to determine if the simulation program is working, since an incorrect switching operation will place the random number limit designation in the wrong position. This event can be detected when the incorrectly switched function block fails and the position specified by the random number does not correspond to that printed out in the sample format.

To check a strategy, several runs are made using different random number sequences. The sample format prints out all the above information for each case. From this information a determination can be made as to whether the simulation is following the rules for the particular strategy.

In addition to performing the function of checking the simulation program, the sample format provides another valuable service. By observing the vicissitudes of the system with respect to the switching patterns which develop, information can be gained about changes in the strategy which might profitably be used to implement more efficient system operation or more economical switching circuitry implementation. This is the manner in which Class γ_2 was derived from class γ_1 .

D. PRODUCTION FORMAT

A typical production run of the computer program simulates system operation for one hundred randomly selected failure patterns. Up to the present time, all runs have included one hundred patterns simply because relatively good estimates of the average system parameters such as total time to fail, number of failures withstood, etc. are obtained without requiring excessive amounts of computer time.

The production format directly provides the following information for each of the one hundred cases:

1. Average time between function block failures
2. Total time to system failure
3. Total number of function block failures before each system failure
(including multiple failures of the same block)
4. Net number of failed function blocks at time of system failure
5. Total number of switching moves experienced by each system
6. Total number of moves made by each spare function block.

In addition to printing out columns of numbers covering the first five items on the list above, most of the data is compiled into bar graphs. Each of these graphs reflects the

performance of the set of one hundred runs with respect to a particular parameter. On the graphs, either discrete points (e.g. net number of failures) or interval terminal points (for continuous parameters such as time) are plotted on the abscissa. The height of the bar above each point or interval shows the number of spares or system simulations which are described by these positions on the abscissa. The program includes a normalization routine for each graph which is used to compute the average, the variance and the standard deviation associated with each graph.

IV. RESULTS

The strategies discussed here (and any new ones which may be invented) must be compared and contrasted to determine their usefulness in increasing the reliability of electronic systems. The primary goal of this comparison is the determination of which strategy provides the greatest net increase in system reliability. Because it appears that the switching circuitry associated with spare blocks increases as the mobility of these blocks increases and because the failure protection effectiveness of added flexibility is non-linear, it cannot be simply assumed that the best strategy is the one with the greatest spare block mobility.

The best way to compare these strategies would be to completely design functionally identical systems using each strategy; get the best available estimates of the failure rates of all the parts; feed this into the computer program and, in the manner described below, plot the reliability versus time curves. The comparison would merely require that one directly observe which strategy has the highest reliability curve. This approach would require a detailed system design for all strategies. To avoid wasting time on strategies which can be shown to be inferior to others with much less detailed input data, several less exact comparisons can be made. These comparisons, which are described below, are the ones which are being made at this point in the study.

A. ^{PERCENT} FAILURES WITHSTOOD (AS_{λ} OF SYSTEM) vs. SPARE MOBILITY

An important consideration in the comparison of systems is the number of failures which can be withstood without system failure. In order to compare strategies with one another where the variable is the number of moves allowed per spare, the number of failures withstood is an important and meaningful criterion. To further compare systems of different sizes on a common base the curves plotted for these systems are expressed in terms of average percent of total system failed versus spare mobility. In figure 5 curves are plotted for three systems of different sizes, 24, 48 and 96 stages employing strategy γ_1 . They are plots of average percent of failures versus number of moves per spare.

These curves provide very useful and interesting results. They are characterized by a sharp rise, a knee and a rapid leveling off. The knee occurs at a small number of moves per spare compared to complete (total system) spare mobility. According to this graph, a great increase in number of failures withstood by a system is effected by increasing spares' mobility up to a point. The increase, then, is diminished and a point is reached beyond which little or no increase in number of failures withstood accompanies an increase in mobility. The characteristic exhibited by these curves illustrates that great increases can be attained in system performance by the introduction of self-repair Class γ_1 with

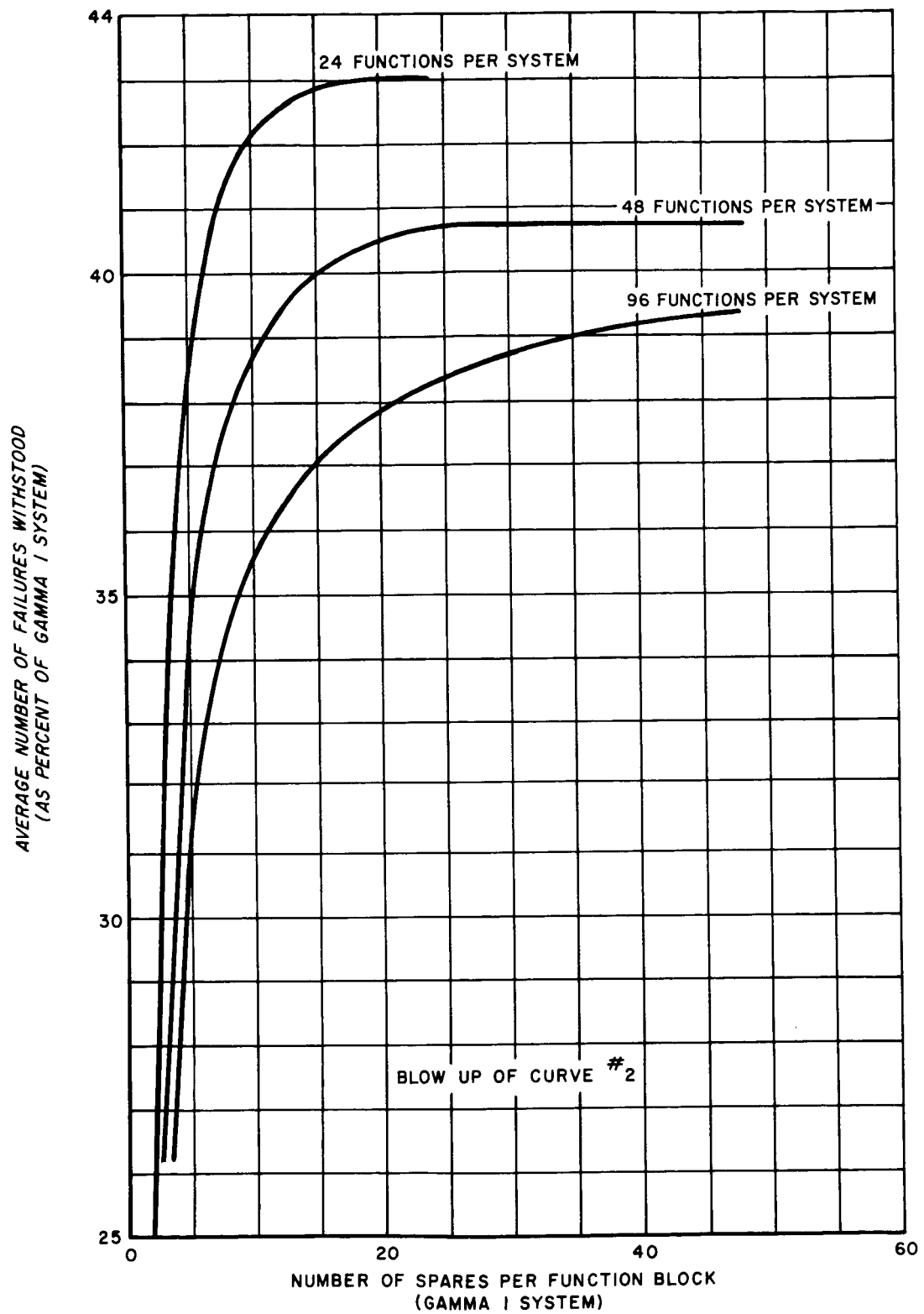


Figure 5. Average Number of Failures Withstood (as Percent of Gamma 1 Systems) Versus Number of Moves Per Spare

relatively little mobility. The addition of more mobility adds little to the effectiveness of the technique. This indicates that the most gain is attained with a small degree of mobility; therefore, the most efficient operation of the technique can probably be accomplished with relatively little switching circuitry.

Plots have also been made for the percent of system failed vs. number of spares per function block for the β class strategies. These plots are illustrated in figure 6. The curves in figure 6 are plots of the Average Number of Failures Sustained versus Number of Spares per Function Block. The results show substantial gains over the multiple-line case for each increase in spare mobility. These curves are restricted to low mobilities because of the fact that the Beta class draws spares to replace failures only from the immediately surrounding area.

Since an important consideration is the worst failure patterns, a plot is shown of the lowest number of failures which were sustained to system failed vs. mobility for the Gamma Class strategies. (See figure 7). These curves agree very closely with those of figure 5 thereby substantiating the conclusion even for the worst case.

Figure 8 shows the Minimum Percentage of Failures Sustained versus Number of Spares per Function Block for the three different length β Class systems. These curves, like those for class Gamma, show a gain over multiple-line system for each advance in mobility.

B. RELIABILITY VS. TIME CURVES

The reliability of a system as a function of time is the probability (P) that the system will be operating correctly at that time, or, out of a given sample, s, P x s of these will be operating correctly. From the production run printout of the computer program, it is possible to plot the percentage of the systems which are operating versus total operating time. This plot closely approximates the reliability curve associated with a particular strategy. The plots made here represent one minus the cumulative sum of the bars of the graph for number of systems failed versus time. For each interval of time in which failures occur a step function is subtracted from the curve corresponding to the number of systems which failed in that interval. This process produces a curve which is a series of discrete steps, starting at 1 and going to 0 as time increases. Smoothing out this curve would result in a curve which is identical in form to the standard s-shaped reliability versus time curve which is common to redundant systems.

As it was mentioned in the introduction to this section, this type curve would be an excellent comparative tool if accurate estimates of the switching circuit failure rates could be made using completed system designs. Because the designs are not yet available, the usefulness of these curves is restricted to that of investigating which strategies are best under

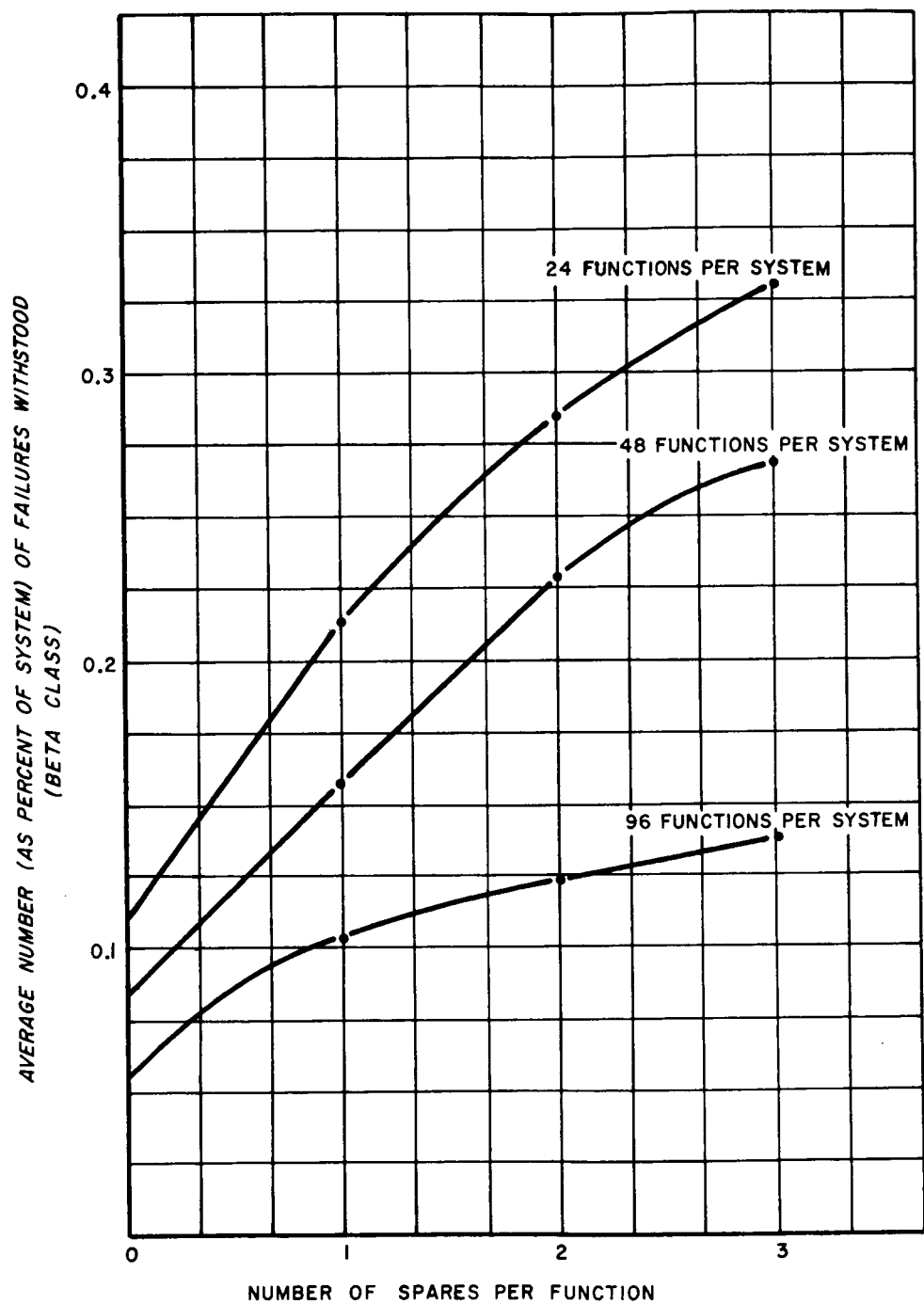


Figure 6. Average Number of Failures Withstood (as Percent of Beta Systems) Versus Number of Spares per Block

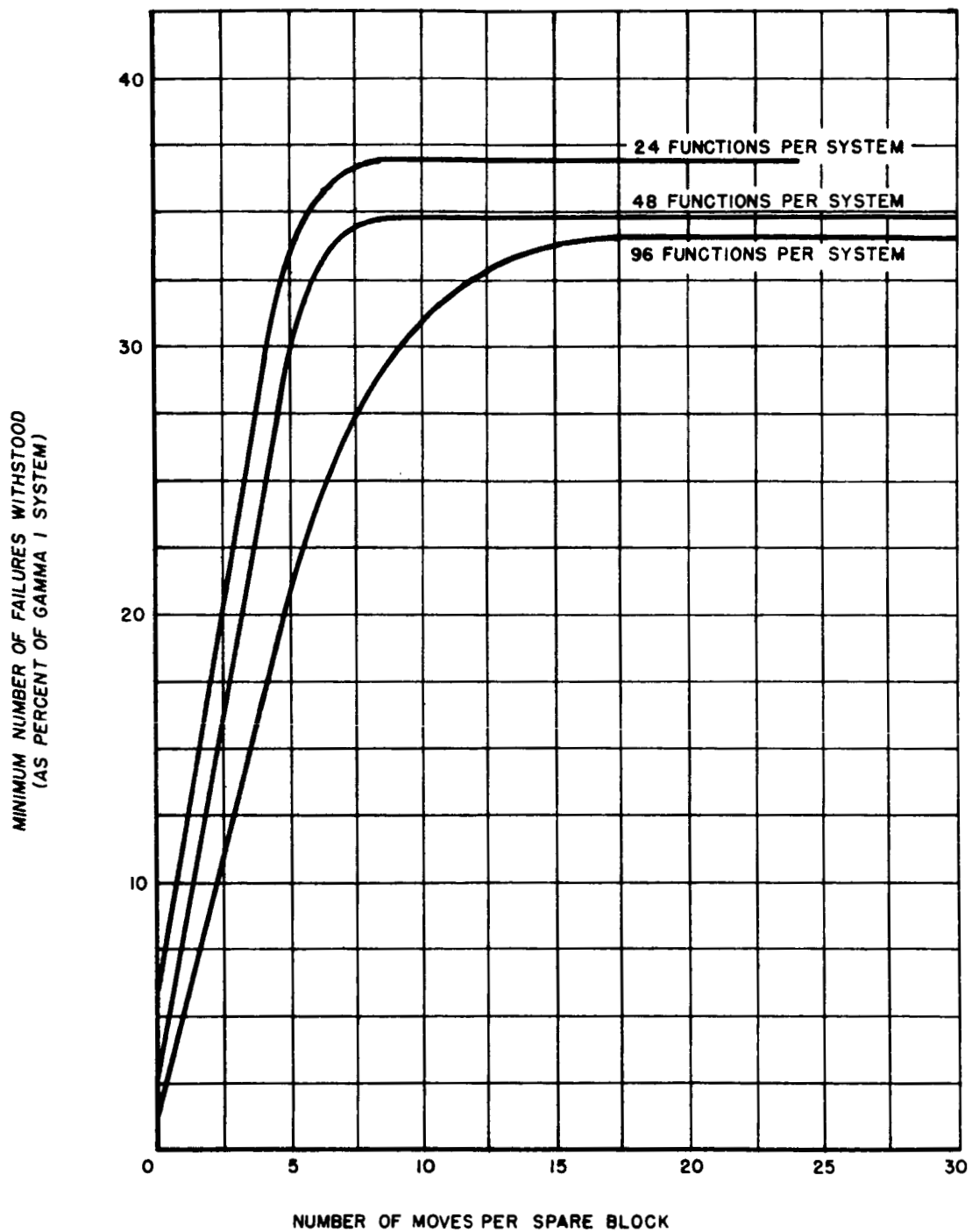


Figure 7. Minimum Number of Failures (As Percent of Gamma 1 Systems)
Versus Number of Moves Per Spare

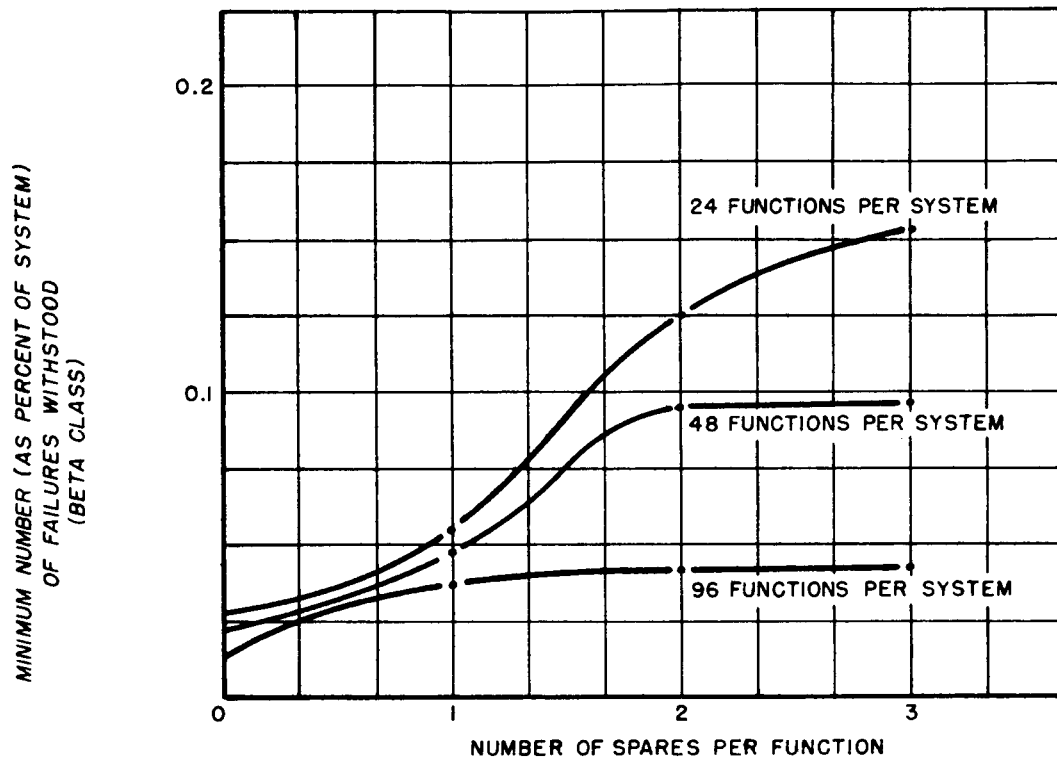


Figure 8. Minimum Number of Failures (As Percent of Beta Systems) Versus Number of Spares Per Block.

certain limiting failure rate conditions. Even under these conditions, the reliability versus time curves are very useful because they provide a universal means of comparing all strategies in all classes.

Examples of these curves for the Beta and Gamma Class strategies are shown in figures 9, and 10. The following comments indicate some of the significant features of these curves.

1. Beta Class Reliability Curves

The reliability curves for the three members of the class are shown in figure 9. The curve for an order-three, multiple-line redundant system is also shown. These curves show a significant gain in reliability of all three strategies of the Beta Class over the redundant case. The effective gain will not be as great in reality because perfect switching has been assumed in plotting the curves.

With the limited amount of switching allowed to strategy β_1 an increase in MTBSF of approximately 100% results. As more switching capability is allowed to the system the reliability continues to increase, showing that strategy β_3 provides significant increase, reliability-wise, over either β_1 or β_2 and very significant increase over the multiple-line redundant case.

2. Gamma Class Reliability Curves

Figure 10 illustrates the reliability curves for four gamma class strategies. Illustrated are the limiting cases 1 move per spare and 23 moves per spare^{*} as well as a multiple-line redundant system. Two strategies of intermediate mobility are also shown.

These curves, again, show that the introduction of a minimal amount of switching capability, 1 move per spare, causes a significant gain in reliability and operating time over the redundant system. It is obvious, also that the first few increases in mobility capability of the Spares induce further noticable gains in reliability over the one move per spare case. As additional mobility is granted to the system, the reliability gained begins to diminish. This is illustrated by the fact that as much gain in reliability is attained by increasing mobility from one to three moves per spare as is gained by going from three to twenty-three moves per spare. This also reflects the flattening effect observed in the curves of percent of Failures Sustained versus Mobility of the System, wherein the additional mobility after a certain point brought no additional gain in reliability.

*24 Function Systems

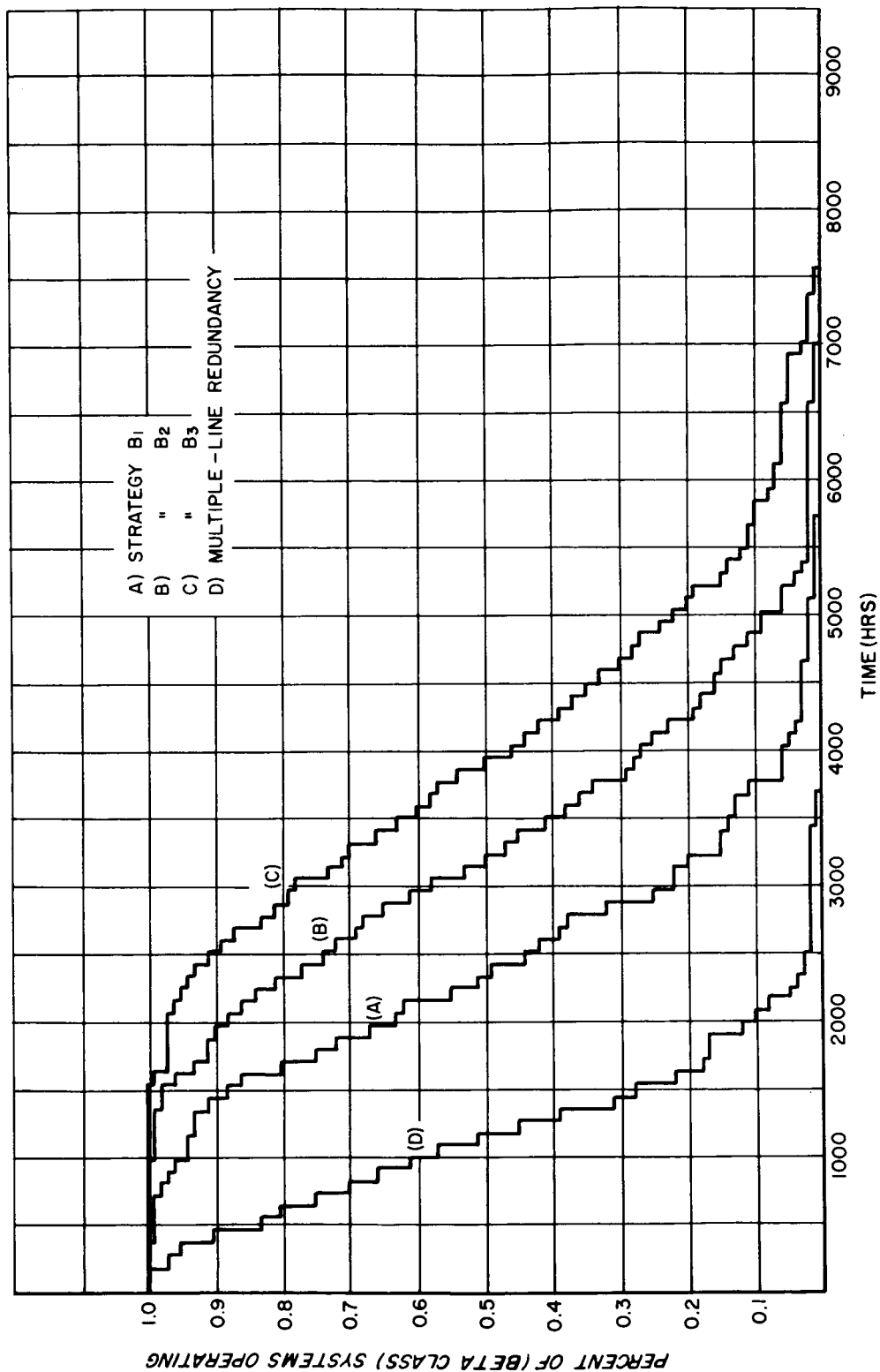


Figure 9. Percent of Systems Operating (Beta Class) Versus Time

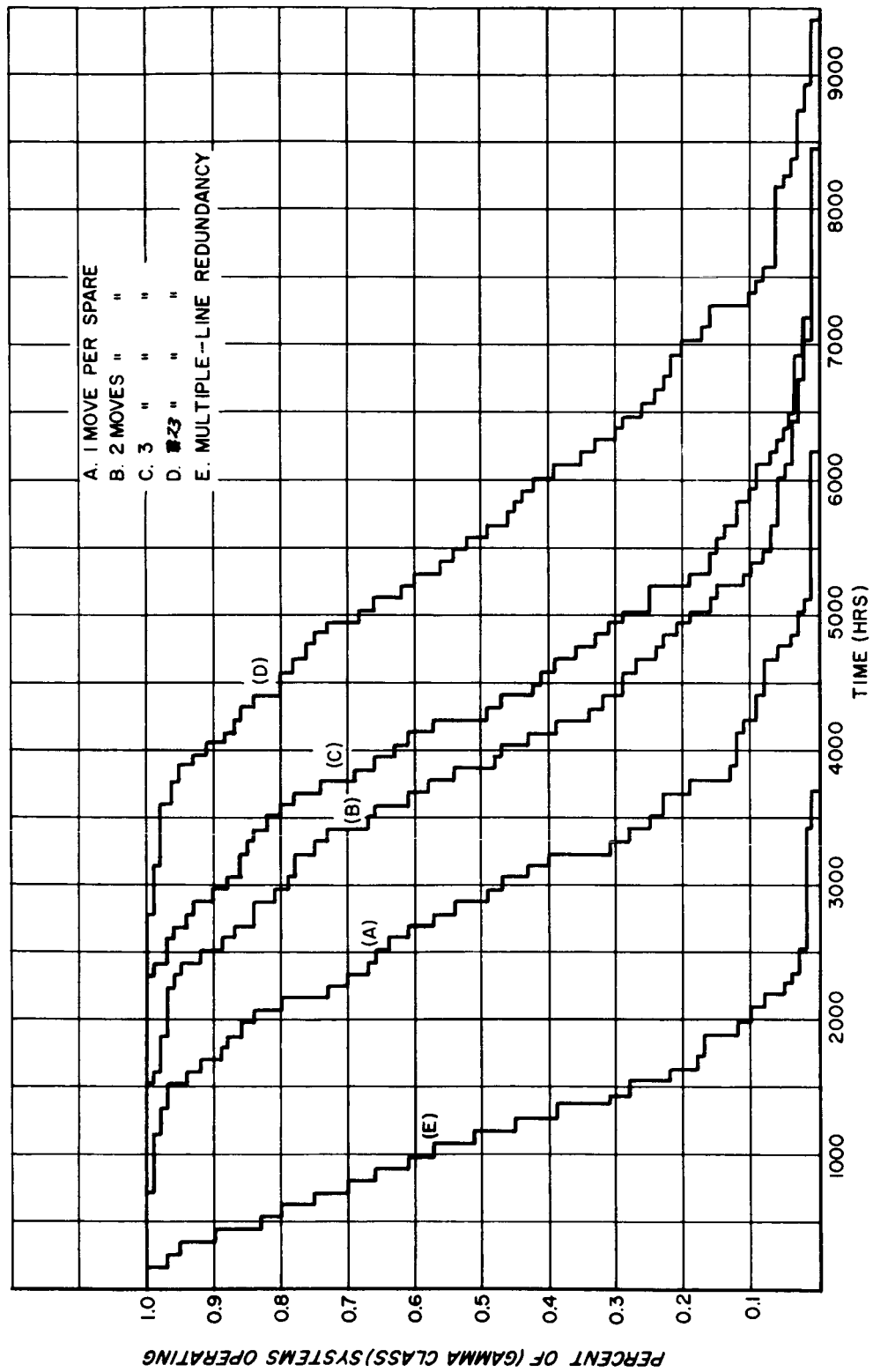


Figure 10. Percent of Systems Operating (Gamma Class 1)
Versus Time

V. SUMMARY AND CONCLUSION

Before self-repairing systems can be implemented, many feasible switching strategies must be considered in an effort to determine the most effective manner to manipulate the redundant or "spare" blocks. The extreme complexity of the reliability expressions associated with these strategies has resulted in the use of a computer simulation program for comparing the effectiveness of the strategies. Rather than proceeding to write separate programs for each strategy, a more general program has been written which employs a small number of subroutines, each of which describes an entire class of strategies. Input data determines which class subroutine is being used and which strategy in a particular class is being simulated. Although this generalized program is a great improvement over the individual program for each strategy approach, it still requires additional programming each time a new class subroutine is added. At this time, the change to a more general program, whose simulation strategy can be completely determined from input data, does not seem to merit the programming time which would be required.

The present program includes subroutines for three classes of switching strategies. Each class subroutine contains a great deal of flexibility, thereby including many individual strategies. This method facilitates easy comparison between members of a class. This comparison allows immediate elimination of many possible strategies as obviously uneconomical. For example, the flattening out of the Percent of System Failed versus Spare Mobility curves (figures 5 through 8) indicate that all possible strategies on the flat part of the curves cannot be optimum strategies.

From the results of the simulation program, curves for Percent of Systems Failed versus Spares Mobility have been plotted for the Gamma Class strategies. These curves have been referenced to that of a multiple-line majority voted system because this particular technique has been the most effective of the passive, failure masking, circuit level redundancy techniques. In all cases these curves show not only that great gains can be realized over multiple-line redundant scheme but that by far the greatest part of these gains are realized for the first few moves allowed to the spare function blocks. Beyond the range of relatively limited mobility, little or no gain in the average number of failures absorbed is realized by the additional mobility allowed to the spares. This is an encouraging result since the great majority of the gain due to self-repair can be retained without the use of an exorbitant amount of switching circuitry.

In the β and γ classes of self-repair strategies the degree of failure masking is the same as that for a multiple-line redundant system of the same order of redundancy. This is due to the fact that no "repair" is made until an ambiguity is present on the output of a

stage. This event corresponds to redundant system failure which activates the switching mechanism and the "repair" is effected. However, until the failure is "repaired" no failure masking is present, and incorrect information may be transmitted to the next stage.

The α class strategies provide additional failure masking because repairs can be initiated by the first occurrence of a failure in any stage. However, because this class implies a higher order of redundancy it cannot be compared to order-three multiple-line redundancy as the β and γ class have been.

The curves of figures 9 and 10 show a very definite gain in reliability for the self-repair strategies over multiple-line redundant systems. The curves for the Beta Class strategies show an increase in reliability for each increase in "repair" capability. Strategy β_3 yields the highest reliability but even strategy β_1 shows a significant gain over the multiple-line system. The reliability curves for the Gamma Class show essentially the same result with respect to the multiple-line case. However, investigation of the curves show that increasing the "repair" capability produces gains for the first few increases after which the magnitude of the gain diminishes. These curves tend to bear out the conclusions drawn from Percent System Failed versus Spares' Mobility curves which flattened out after a certain mobility was reached. The gains illustrated here must be considered as ideal because the switching circuitry for self-repair is here assumed to be perfectly reliable. More realistically, the gains obtainable will be a function of the switching circuitry complexity and will not be as great as shown here.

VI. FUTURE STUDIES

All of the computer simulation results discussed in this report have been based on the assumption that the switching circuitry was perfectly reliable. Efforts are now being made to determine the range of allowable failure rates which can be associated with each strategy for it to be of maximum effectiveness. These ranges are to be studied as a function of the failure rates of the associated signal processor blocks. As a result, before actual system designs are begun, information specifying the optimum switching strategy corresponding a given signal processor failure rate should be available.

From the sample and production simulation run printouts it has become obvious that many of the spare function blocks do not experience as many switching operations as they have the capability for. When all spares are assigned a uniform mobility some reach their limit and, in doing so substantially extend the life of the system. However, in many cases when system failure has occurred, there are many spares remaining which have not been used to any great extent. In order to capitalize on this phenomenon a class of strategies γ_2 is being developed which will assign different mobilities to the spares in a stage. Class γ_2 will be simulated by a new sub-routine which is being written for the computer program. When data is available comparisons will be made between this and the other classes. Additional classes will be simulated in a similar manner as they are developed.

None of the strategies considered so far have permitted spares to return to previous locations. It is possible that removal of this restriction might add to the failure absorption capability of a system. This area certainly should be explored in this study series.

Although little has been said about the physical switching techniques to be employed, it has been tacitly assumed that the failure detection and replacement circuitry would be combined as much as possible. It has been suggested that these two phases of the repair function might profitably be separated and made almost completely independent from a circuit viewpoint. This is another area which should be given careful attention.

The Alpha class strategies have not been thoroughly investigated to determine the optimum degree of spare overlap (i. e. , two sets of spares serving some of the same functional region). The information from this investigation should influence the design of new strategy classes as well as indicating the optimum strategy for the Alpha class.

VII. APPENDIX

A. CLASS α

Illustrated in figure A-1 is an α class strategy wherein each spare can "repair" failures in one row and either of two stages. Spare "1" can "repair" stages 1 or 2; "2" can "repair" 3 or 4, etc. Each spare can repair failures only in its own rows. This can be expanded such that, for example, three spares can each repair function blocks in any of ten stages or, in general, r spares for n stages. Overlapping of spares capability may help guard against "lumped" failures.

Many different strategies and system repair capabilities can be developed by simply varying r and n or by overlapping possible individual spare "repair" ranges.

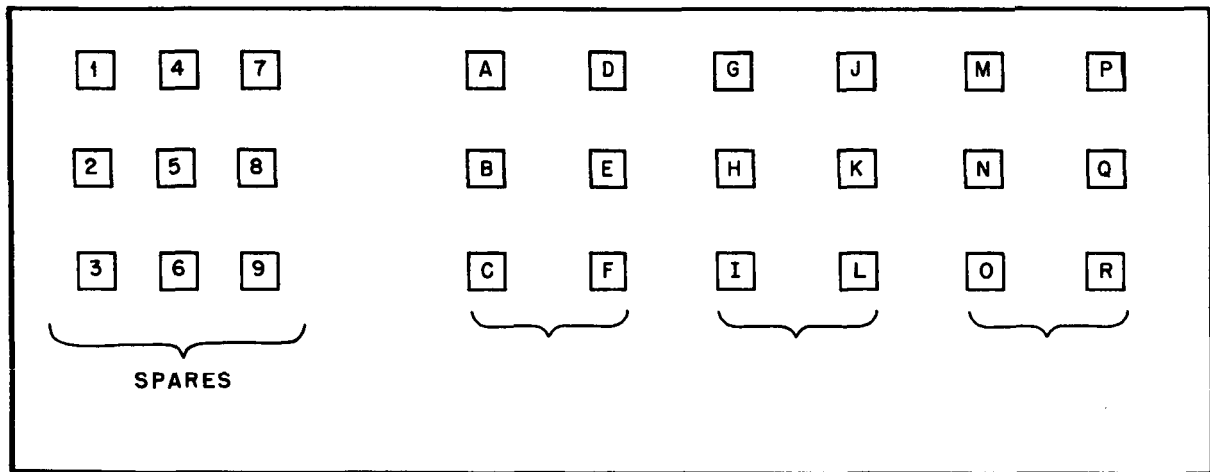


Figure A-1. Alpha Class Self-Repair

B. CLASS β

There are presently three specific strategies of β Class. The major difference between these strategies is the number of spare function blocks which can replace a given failure.

1. Class β_1 (Figure A-2)

Class β_1 allows only one "spare" for a given failure response. For example, function block "H" is given capability as a spare for stage # 4. Figure A-2a shows the system before failures occur. When one function block, J, in stage #4 fails no switching results other than the elimination of the failure. (See figure A-2b). When the second failure, say K, occurs in stage #4, function block "H" will move into stage #4 (See figure A-2c.) and resolve the ambiguity caused by the failure. After the failed block has been eliminated block "H" remains in stage #4.

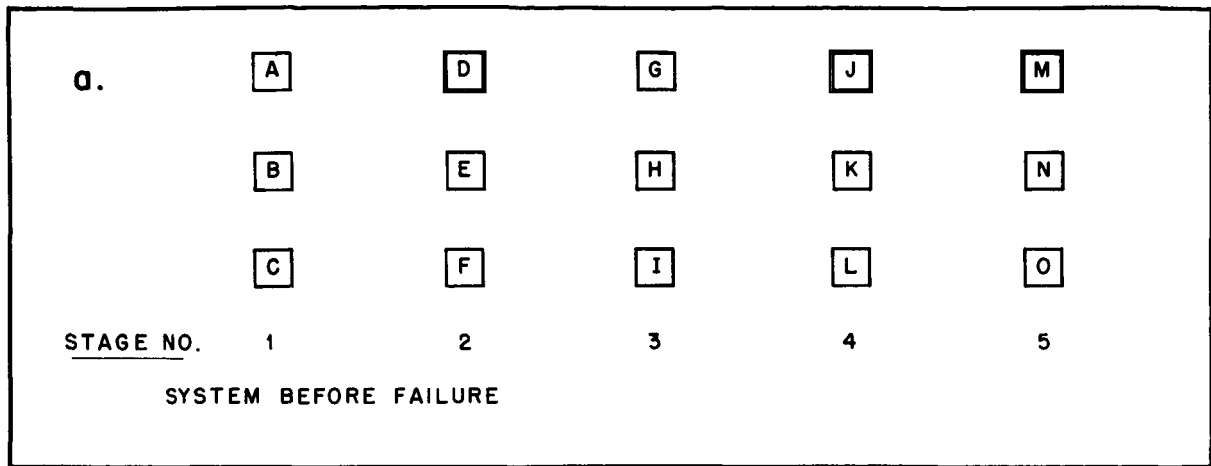


Figure A-2a. Beta Class Self-Repair

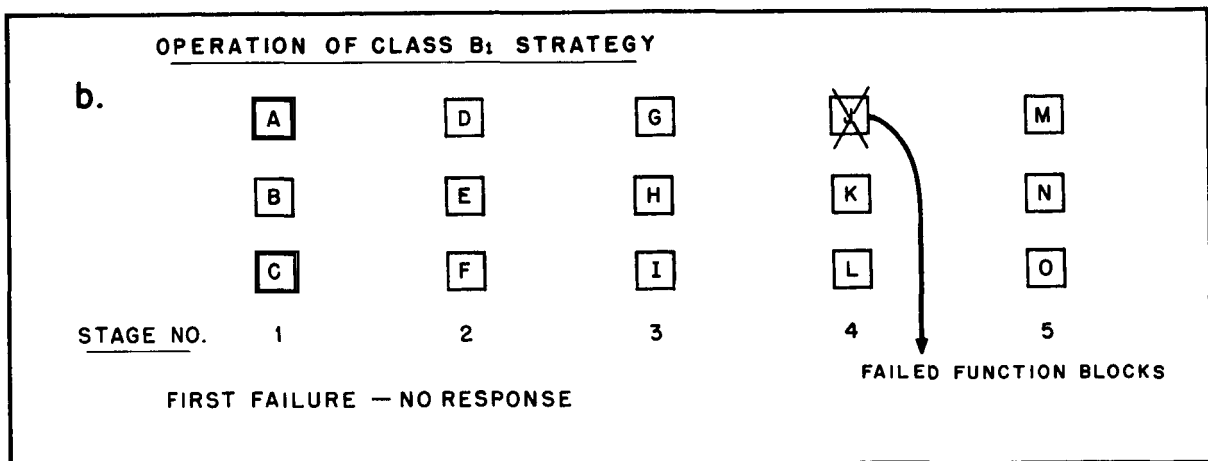


Figure A-2b. First Failure

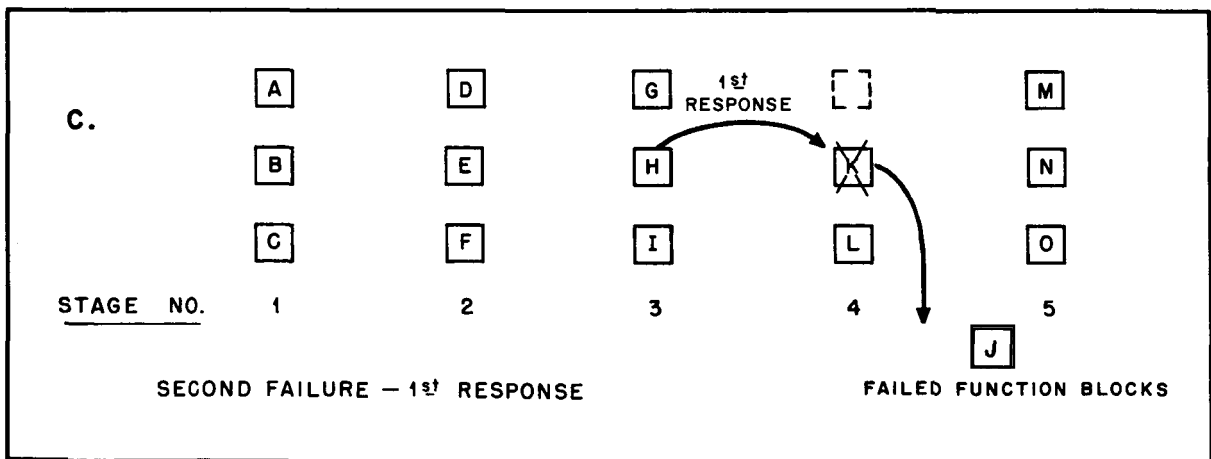


Figure A-2c. Second Failure Response

It is possible that one function block will remain working alone without system failure. For example, if function block "G" failed before "K" function block "I" will carry the load for stage 2 after "H" switches until it fails. (See figure A-3.) System failures occur when a lone operating function in a stage fails or when no spare is available to resolve an ambiguity. Failure of this system could occur when function block "E" and "G" have failed and failure of blocks "H" or "I" occurs (figure A-4), since for this strategy, block "E" is the only spare capable of "repairing" a failure in stage #3.

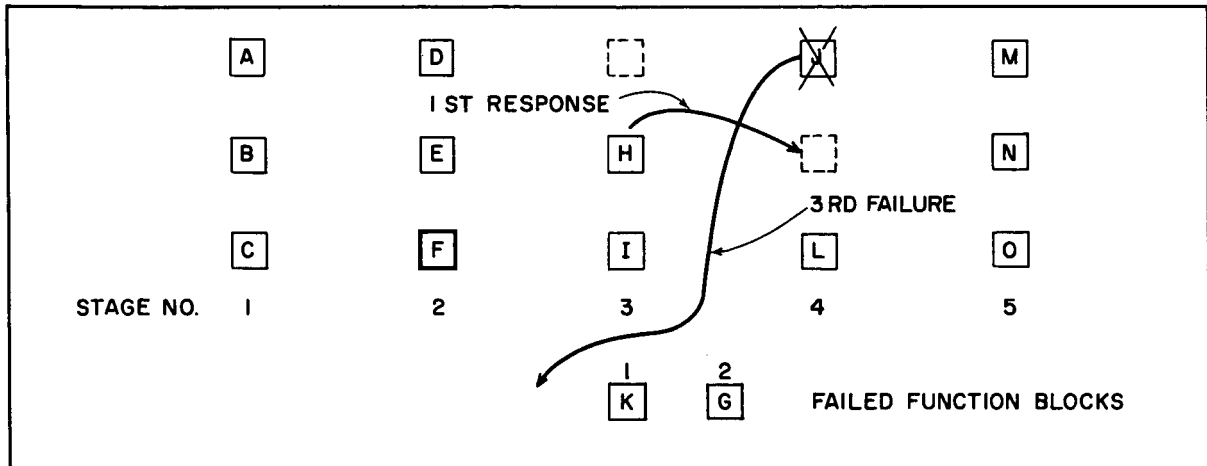


Figure A-3. Third Failure Response

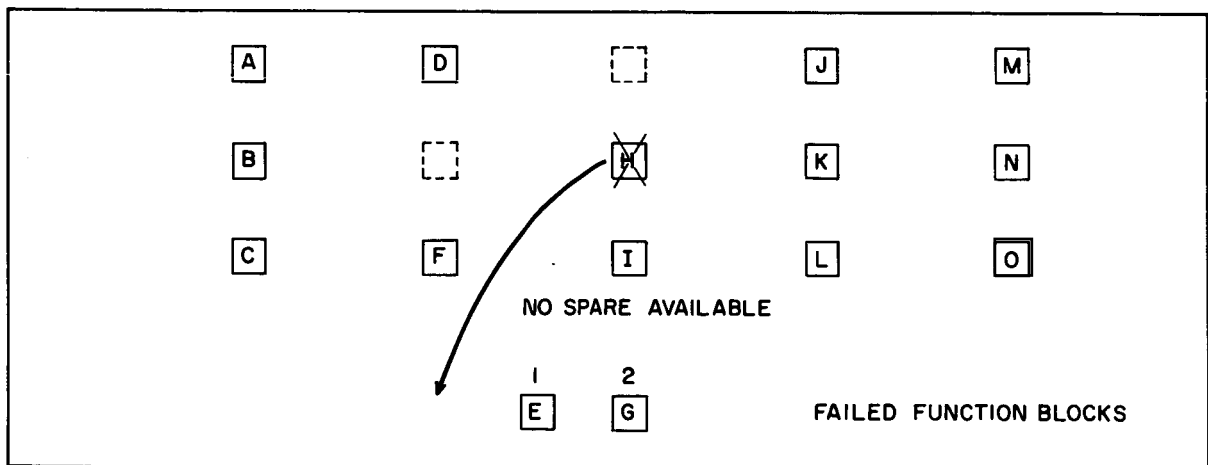


Figure A-4. Catastrophic Failure Sequence

2. Strategy β_2 (Figure A-5)

Strategy β_2 is similar to β_1 , but it allows one additional function block to replace failures in a given stage. In strategy β_2 function block "M" in addition to "H" is given the capability of replacing failed blocks in stage #4. Strategies β_1 and β_2 operate

identically through the first two failures. When the third failure in stage #4 occurs block "M", if still operative, will switch into stage #4 in the same fashion as did function block "H" in Class β_1 . This move is labeled "2nd response" in figure A-5. System failure in strategy β_2 occurs in the same manner and under the same conditions as in strategy β_1 .

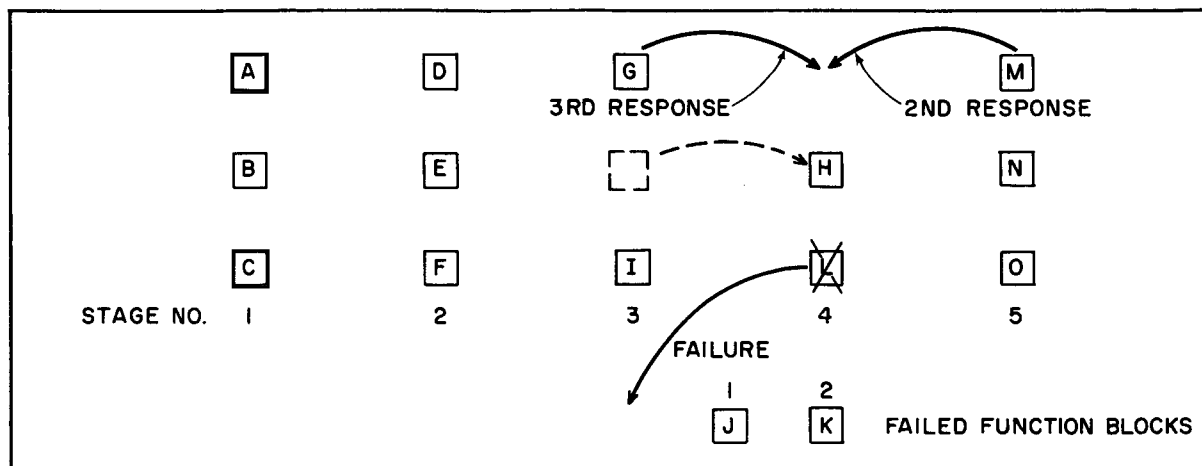


Figure A-5. Beta 2 & 3 Strategy

3. Strategy β_3 (Figure A-5)

Strategy β_3 extends the scheme one step further. Here, a third function block is allowed to move in addition to the two responses allowed to strategy β_2 . In this strategy the ability is imparted to function block "G" in stage 3 to replace failed blocks in stage #4. This is the 3rd response shown in Figure A-5. Again, failure occurs in the identical fashion to the other two strategies.

C. GAMMA (γ) CLASS

Gamma Class is divided into two parts: Class γ_1 , where all spare function blocks have the same mobility, and Class γ_2 where one spare in each stage has a greater mobility than the other.

1. Class γ_1 (Figure A-6)

As in Beta Class strategies, the first failure in a stage of a Gamma Class system evokes no response from the system. The second failure creates an ambiguity on the output of the stage. This activates the switching mechanism to switch block "H" into stage 4 thereby dissolving the ambiguity. (See Figure A-5b.) The second failed block is now identified and switched out of the system. Block "H" remains in stage 4 to detect subsequent errors. another failure occurs in stage 4, for example block "L", block "G" from stage 3 will switch into stage 4 in the same manner as did block "H". This leaves no error detecting capability in stage 2. To overcome this, block E from stage 2 switches into stage 3 to fill the void created by the switch of block "G". (See figure A-6c.)

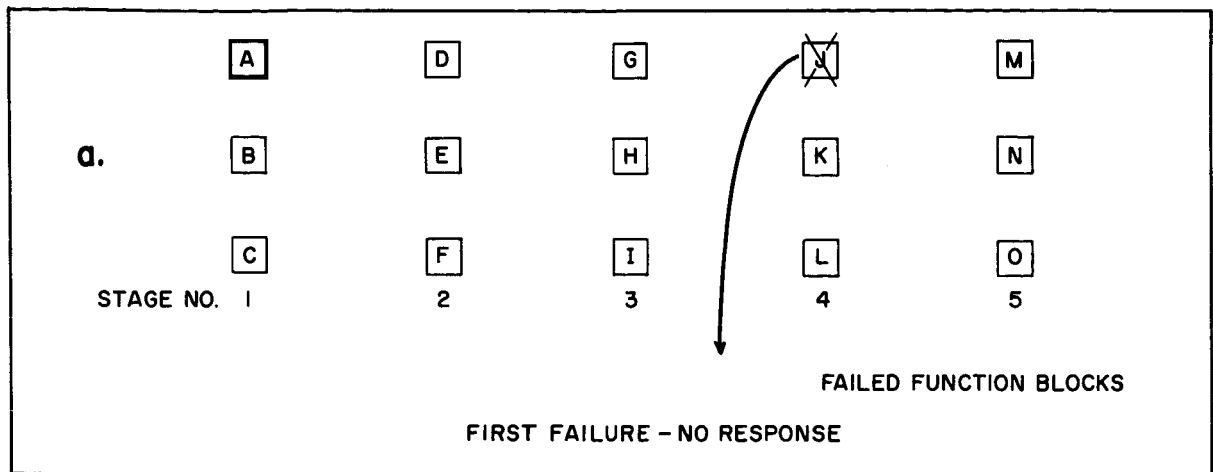


Figure A-6a. Gamma 1 Strategy - First Failure

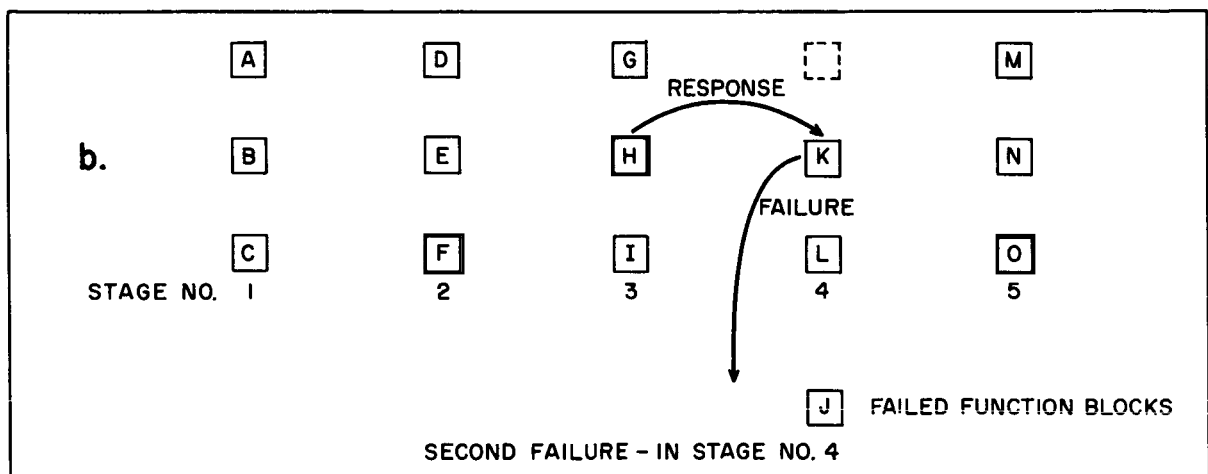


Figure A-6b. Second Failure Response

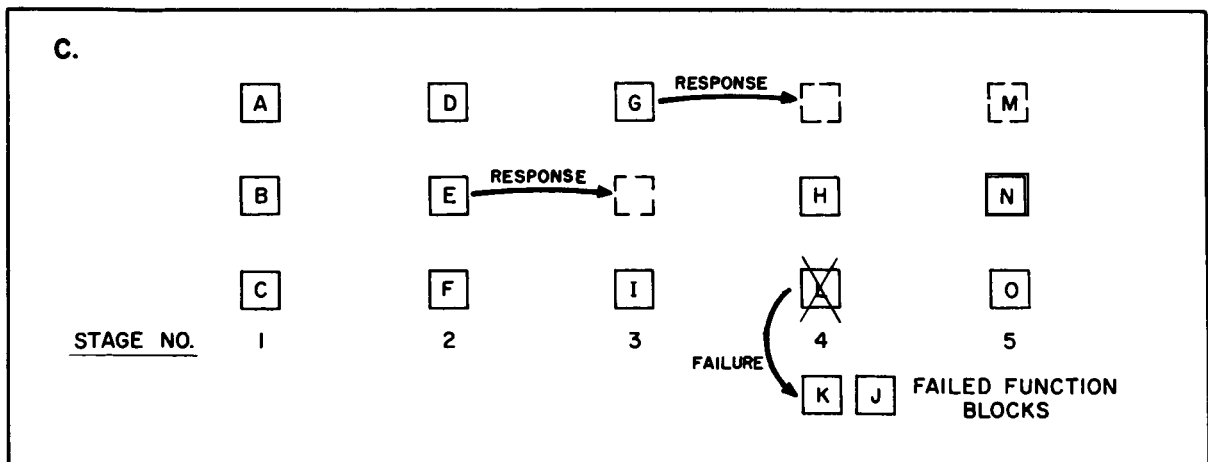


Figure A-6c. Third Failure Response

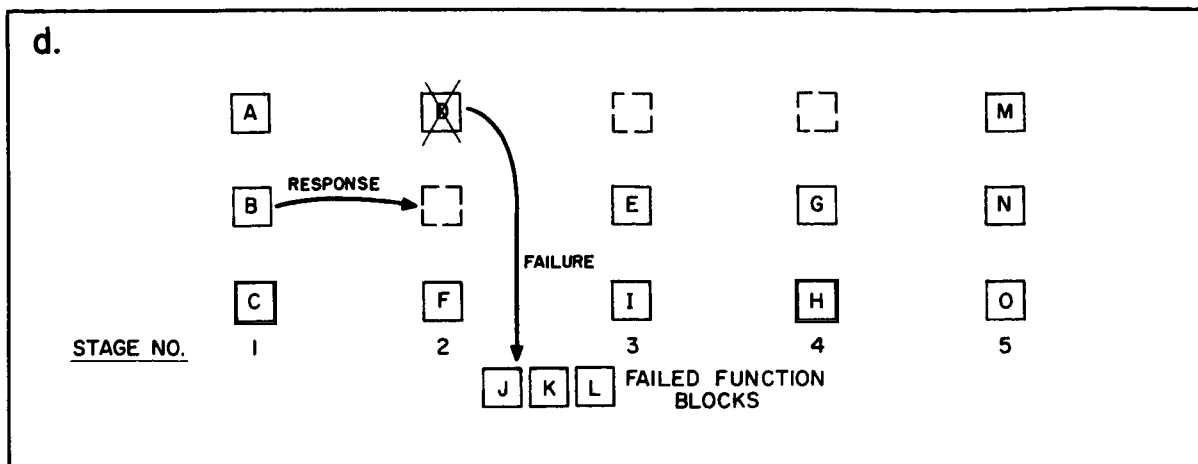


Figure A-6d. Single Block Operation

Now if a failure should occur in stage 2, block "D"; a spare function block "B", from stage 1 will switch to stage 2 and the failed block "D" will be switched from the system. (See figure A-6d.) As additional failures are sustained this process continues until a limit is reached. The end to this process can be reached in one of two ways:

1) A limit can be set for the mobility of a particular function block. In this case, once a function block has reached its limit it can no longer act as a spare for failures in the stage following it. If a critical failure occurs and all possible spares have failed or reached their limits the system fails. Voids which cannot be filled due to spares reaching their limit remain as voids but the system continues to operate until the remaining function block fails. This limit sequence is illustrated in figure A-7a. Block "A" has a

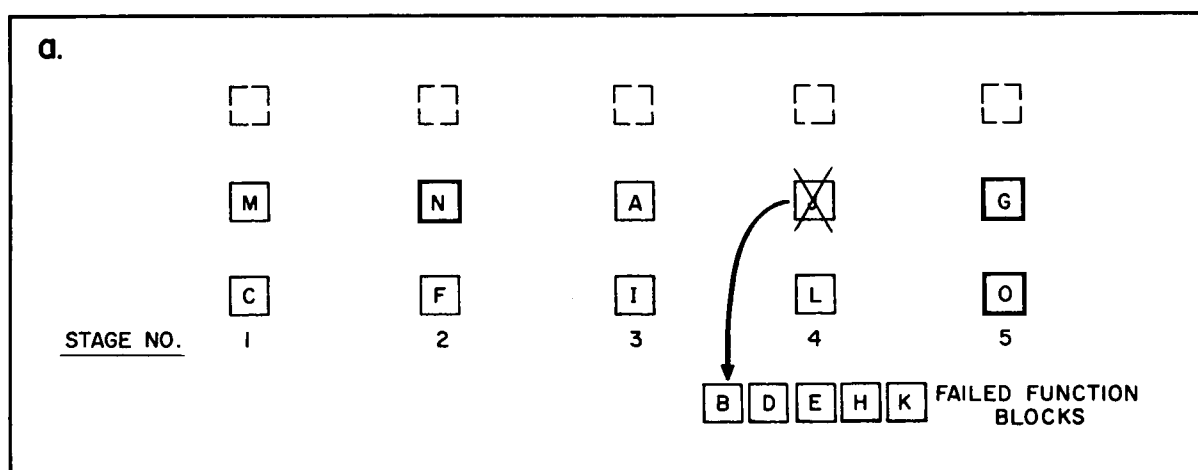


Figure A-7a. Function Block Limit

mobility of 3 and after a given failure pattern the system appears as in Figure 7a. Block "A" has reached its limit. Upon the occurrence of a critical failure in stage #4, block "A" cannot act as a spare for this stage. The ambiguity remains on the output of stage 1 and the system is considered failed. However, if the critical failure occurred in stage 2 rather than stage 4, block "M", since it hasn't reached its limit, would switch into stage 2 and resolve the ambiguity. This leaves a void in stage 1. Function block "G" cannot switch into stage 1, hence, the void remains and the system works properly as long as the remaining block in stage 1 does not fail.

2) Another failure mechanism can exist for class γ . When the system has sustained a large number of failures such that the number of remaining spares is equal to the number of stages this second mechanism case becomes effective. When an additional failure occurs, each spare function block will respond once, the initial one will resolve the ambiguity and others will fill the successive voids which appear in the immediately preceding stages. Since there is now one less spare than there are stages a void must remain somewhere in the system. If the next failure is in the stage which contains the void or that stage for which the void would have been a spare, the system goes down. For example, referring to Figure A-7b if function block "G" fails, block "D" will switch into #4 to correct for the failure. Block "A" will fill the void for block "D", block "M" for "A" and block "H" for block "M". The process stops here. There is a void in stage 5. Now failure in stage 1 or stage 5 will cause system failure. Class γ_1 , allows uniform mobility to each spare function block in the system.

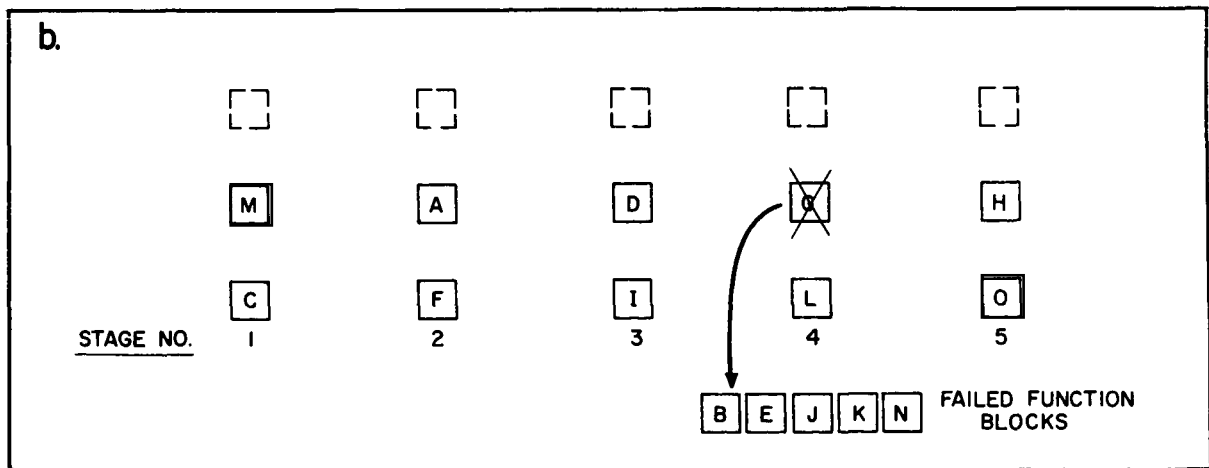


Figure A-7b. Marginal Operation

Many different strategies are contained under the heading of Class γ_1 . These differ primarily in the limit assigned to the mobility of the spare function blocks. A particular strategy may be identified by specifying "n" in the statement "n moves per spare." The value of n prescribes where a given function block will reach its limit and therefore controls the differences between the various strategies of Class γ_1 .

2. Class γ_2

Unlike the Gamma 1 Class, which assigns the same mobility to all spare function blocks, Gamma 2 Class allows the two spare function blocks to differ from one another in mobility. Figure A-8 will assist in the description of the switching processes which occur for strategy Gamma 2. The members of the top row are assigned a mobility 3, those of the middle row, a mobility 2.

The first failure in a stage will evoke no response aside from the elimination of the failed block from the system. Upon failure of the second function block in a stage (stage 4), the spare will be drawn from the next stage (stage 3). Block "G" which has the greater mobility will switch from stage 3, to stage 4. (See figure A-8a) This is the only switch which will

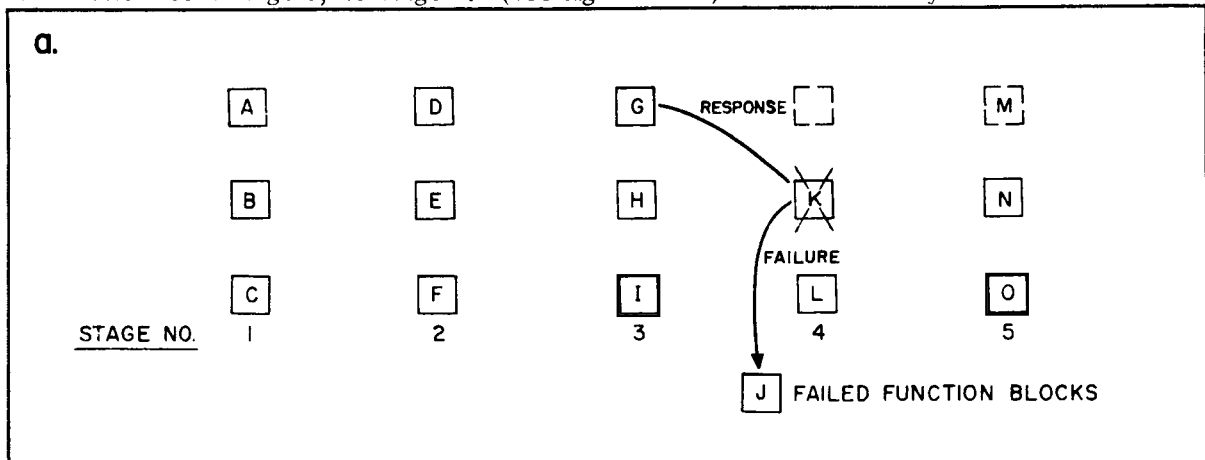


Figure A-8a. Gamma 2 Strategy - First Failure

occur. Since there are two function blocks remaining in stage 3 the void created by the switch will not be filled. The next failure occurring in stage 4 will require another spare to be switched into the stage. This spare is drawn from next stage which has a spare with high mobility and which is within range to supply the need i. e., block D from stage 2 will switch into stage 4. (See figure A-8b.) This leaves another void which is not filled and which needs not be filled. In the system described in figure A-8, the next failure in stage 4, cannot

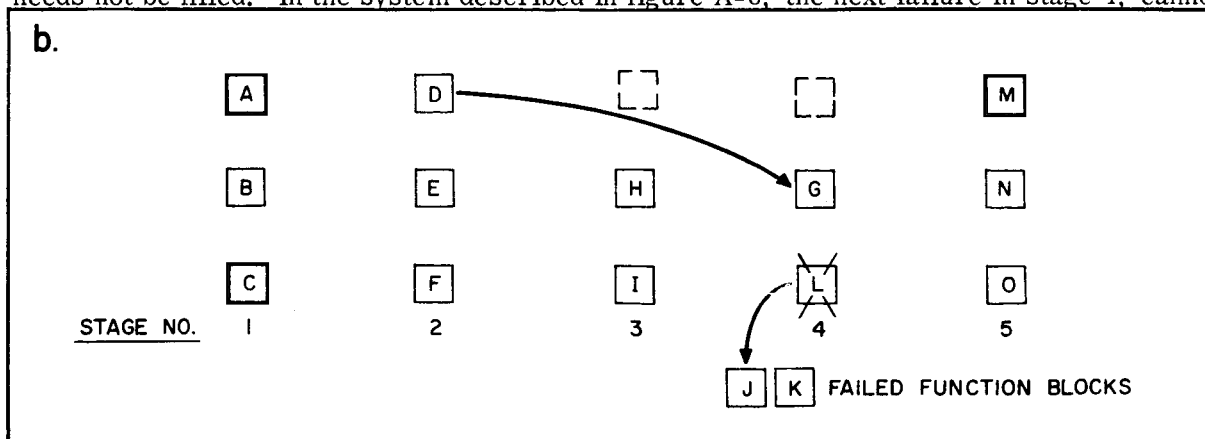


Figure A-8b. Gamma 2 Strategy

draw a high mobility spare A, because it is out of range for stage 4. In this case the lower mobility spare from stage 3 is used spare "H". This leaves a void in stage 2 which must be filled since there is only one remaining operating function block in that stage. This void is filled as though it were a failure; if a high mobility spare is available it will be switched, i. e., function block "A" will switch to stage 3. (See figure A-8c.) This process continues until either a failure occurs and no spare is available or a lone remaining function block in a stage fails. System failure occurs at this point.

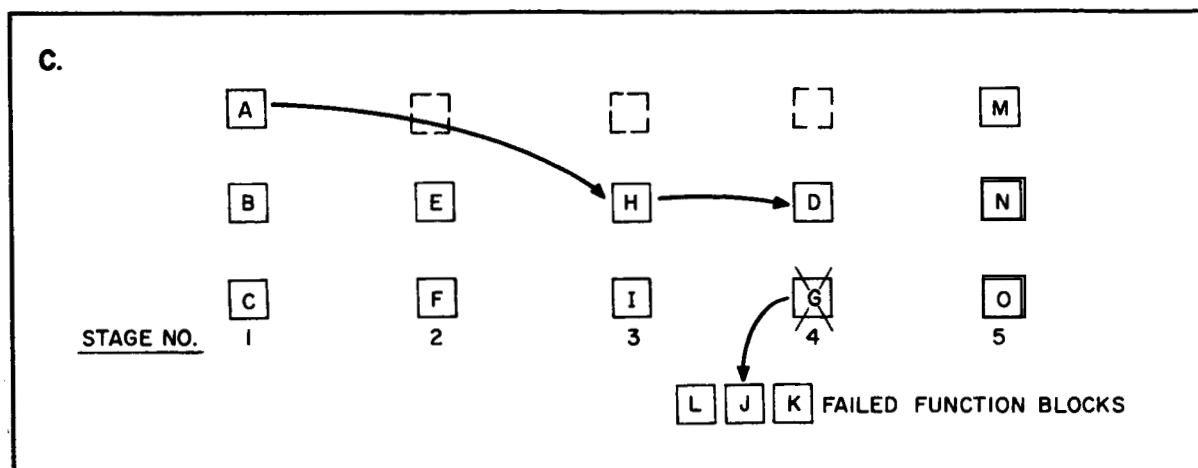


Figure A-8c. Gamma 3 Strategy - Third Failure Response